Amendments to the Drawings

The attached sheet of drawings includes changes to Figure 2. This sheet replaces the original sheet of drawings.

Attachment: Replacement sheet.

Remarks

Applicants respectfully request reconsideration of the present application in view of the foregoing amendments and the following remarks. Claims 1-17 are pending in the application. Claims 1-17 are rejected. No claims have been allowed. Claims 1, 10, and 13 are independent. Editorial amendments have been made to claims 1-3, 5, 10, and 13. No new matter has been added.

Cited Art

The Action cites U.S. Pat. No. 6,560,720 to Chirashnya et al. ("Chirashnya"); U.S. Pat. No. 7,013,482 to Krumel ("Krumel"); Kim et al., "Design and Implementation of Home Network Systems Using UPnP Middleware for Networked Appliances", IEEE Transactions on Consumer Electronics, Volume 48, Issue 4, Nov 2002, pages 963-972 ("Kim"); Dugan et al "Design of Interfaces for Power Systems Analysis Components", Power Engineering Society Summer Meeting, Volume 2, 18-22, Pages 852-857, July 1999 ("Dugan").

Drawings

The Action objects to the drawings as failing to comply with 37 C.F.R. 1.84(p)(5) for including a reference "200" which is not mentioned in the description. With the above amendment to the drawings, which replaces Figure 2, the extraneous reference is removed. Applicants therefore request that the objection to the drawings be withdrawn.

Claim Rejections - 35 USC § 112

The Action states that "[c]laims 1-9 and 14 are rejected under 35 U.S.C. § 112." [Action, at page 3, para. 8.] However, the Action then proceeds to withdraw the previous rejections of claims 1, 5, and 14 under 35 U.S.C. § 112 based on the previous Amendment filed on January 5, 2007. [*See*, Action, at pages 3-4, paras. 9-11.] Applicants thus assume that the statement of rejection in paragraph 8 on page 3 is due to a typographical error. If Applicants are mistaken, Applicants request that the Examiner explain the rejection in a future communication.

Claims Rejections - 35 USC § 102

The Action rejects claims 1-5, 13-16 under 35 USC § 102(a) as being anticipated by Kim. For a 102(a) rejection to be proper, the cited art must show each and every element as set forth in a claim. (*See* MPEP § 2131.01.) However, the cited art does not describe each and every element. Accordingly, applicants request that the rejection be withdrawn.

Claim 1

Claim 1 recites, in part:

> A method of generically emulating devices in a device connectivity protocol, the method comprising:
>
> *processing, in an device emulator, a description of a device to be emulated* in the device connectivity protocol, the description specifying a set of actions of the device to be emulated;
>
> in response to receiving an action request at the device emulator per the device connectivity protocol, *validating, in the device emulator, to which action out of the set of actions specified in the description the action request matches;*
>
> upon validating an action to which the action request matches, *producing, at the device emulator, a default response, the response based on the description such that, through the response the device emulator emulates operation of the device to be emulated.*

[Emphasis added.]

For example, the Application, describes the utility of generic device emulation:

> In emulating a device based on its description, the generic device emulator provides default behaviors for a set of capabilities defined in the description (e.g., for a set of actions defined in UPnP™ service descriptions for services of the device). . . . The device developer therefore need not provide specific behavior implementations of the device's capabilities, and the generic device emulator still provides an emulation of the device operating within the protocol meeting the device's definition. Further, the device developer can provide specific behavior implementations of none, some or all of the device's capabilities, as desired. . . . The generic device emulator supplies the default behavior of those capabilities for which no specific behavior implementation is provided.

The Application then goes on to describe more specific examples of device emulation:

> With reference now to Figure 2, *the generic device emulator 210 emulates the operation of an emulated device (e.g., devices 130-132) within the network architecture 100 (Figure 1) of UPnP™, including the behavior of the emulated device* during Addressing, Discovery, Description, Control and Eventing phases of UPnP™. *In other*

> *words, the generic device emulator effectively provides an implementation of the emulated device as it would operate within the UPnP™ protocol. . . .*
>
> The generic device emulator 210 emulates the behaviors of the emulated device within UPnP™ based on the UPnP™ description of the device . . . . Given the device and service descriptions 220-221 of any UPnP™ device, the generic device emulator 210 emulates behaviors implementing the description within UPnP™. . . .
>
> In the control phase, the generic device emulator 210 implements default behaviors for the actions specified in the emulated device's service description document(s) 221. The generic device emulator 210 receives action invocation messages (SOAP commands) from control points and validates these messages against the emulated device's service description. *Upon validation, the generic device emulator 210 provides a default response to the action invocation, which response conforms to the data format and types specified in the service description for the action. For example, if the response to the action specified in the service description is to return an integer value, the default behavior simply returns a default integer value (e.g., a zero).*

[Application, at page 13, line 2, to page 14, line 2.] The application goes on to discuss the acquisition of a device description at page 16:

> Device and Service Info
>
> Once a device description document 220 is given to the emulator 210, the device info and all the service info are parsed and are maintained by the Device and Service Info object 320. The object 320 provides methods by which the other components of the emulator can get information about the device and its services.

[Application, at page 16, lines 20-24.]

The Action cites to sections of Kim which describe the operation of a home server which only displays device info and allows actions to be sent to devices, rather than an emulation device. As such, these sections do not teach or suggest "producing, at the emulator device, a default response, the response based on the description such that, through the response the emulator device emulates operation of the device to be emulated." In its rejection of this language of claim 1, the Action cites to page 965, column 1 and Figures 2 and 10 of Kim. Page 965, column 1 of Kim, however, is focused on actions of the home server:

> The home server checks all the UPnP-compatible appliances in the home network . . . . When a UPnP-compatible appliance is found, a display icon of the appliance and its name appear in the tree view . . . . In the appliance section of the home server program display, simple information about each monitored appliance (the state variables) is displayed at the top. When the user invokes an action service using

> the action item in the device control frame, a message is sent to invoke the action. When successful, the updated state variables are displayed on the home server.

[Kim, at page 965, column 1, paragraphs 1-4.] Applicants note as well that cited Figure 2 is a "Block Diagram of the Home Server Program." Therefore, while the Figure does describe an "Action Invocation" loop this is all done by a home server. Furthermore, as this passage demonstrates, the home server description of page 965 column 1 is clearly focused on using the home server to *send messages to appliances* in order to invoke actions in the appliances and does not describe either emulation nor a "response." Thus, Kim's home server cannot teach or suggest the above-quoted language of claim 1.

Applicants also note that Kim utilizes Appliance Emulators, which are separate from Kim's home server:

> Home appliance emulators can exchange many types of messages or control data. The home server sends a discovery request to a newly connected appliance emulator and the appliance responds.

[Kim, at page 968, column 1, paragraph 2.] Kim's description of the home server sending messages to a "newly connected" appliance emulator demonstrates that the two entities are distinct. Thus, as Kim's home server is separate from the appliance emulators, and for the reasons above, it does not perform any emulation, and cannot "produc[e], at the emulator device, a default response, the response based on the description such that, through the response the emulator device emulates operation of the device to be emulated," as is recited in claim 1.

Kim's "appliance emulators" are each built for a specific device emulation job and do not utilize a description to operate. Therefore they do not teach or suggest various "description" language of claim 1. Kim does describe "Appliance Emulators" However, Kim does not give much in the way of particulars about how they are implemented. Applicants note that, instead, Kim demonstrates examples of various appliance emulators at Table 2 and Figure 9. With regard to Table 2, Kim describes emulators based on different operating systems (e.g., the "O.S." column) and with differing display types (e.g., the "display type" column). [Kim, at Table 2.] This demonstrates, at the least, that Kim's appliance emulators were likely built separately, at least because they were built in different operating systems. This fact is supported by Figure 9, which shows four widely different user interfaces for four different appliance emulators.

These differences make it difficult for Kim to demonstrate generic emulation. It would be unlikely, if Kim appliance emulators were able to teach or suggest "processing, in an emulator device, a description of a device to be emulated" and "producing, at the emulator device, a default response, the response based on the description" that different devices would be emulated on different platforms and with different GUIs. Indeed, Kim's teaching of disparate emulations would teach away from the utility of using a device description.

The Action, however, notes Figure 10 from Kim, which shows XML descriptions of two appliances. [*See*, Action, at page 5, paragraph 14.] While Figure 10 does appear to show descriptions of appliances, namely that of an air conditioner and a toaster, there is no indication in Kim that these descriptions are either "processed," "validated," by the appliance emulators of Kim, nor are they used to emulate devices. In particular, the only passage of Kim describing the use of XML files reads:

> When an air conditioner emulator is plugged into the network, it sends device information to advertise its existence. Then, *after the home server receives information about the home appliance using the XML page, the action service is activated.*

[Kim, at page 968, column 1, paragraph 3.]

Through this passage, Kim describes the *sending* of an XML description from an appliance emulator to a home server. While this might imply that the application emulator has a copy of the file, it does not demonstrate any teaching or suggestion in Kim that the appliance emulator "processes," or "validates" the file. Furthermore, as Kim does not describe any particular actions performed by the appliance emulator with regard to an XML description beyond sending the file, Kim does not teach or suggest "producing, at the emulator device, a default response, the response based on the description," as it is not clear that Kim's appliance emulators have any detailed knowledge of the description.

For at least these reasons, Kim cannot teach or suggest each and every element in claim 1. Thus, the rejection of claim 1 over Kim is improper. Applicants therefore request that the rejection of claim 1, and of dependent claims 2-5, be withdrawn and claims 1-5 be allowed.

Claim 13

Claim 13 recites, in part:

> *Computer-readable media having stored thereon a software framework of a generic device emulator for execution on a computer to provide emulation of an operation of a device within a device connectivity architecture consistent with a textual description of the device*, . . . the generic device emulator comprising:

. . .

program code for performing a default behavior producing a response for the action consistent with the data format specified in the description , thereby emulating operation of the device for the action.

In its rejection of the quoted language in claim 13, the Action cites to the same portions of Kim as were discussed above with respect to claim 1. Thus, for at least the reasons discussed above with respect to claim 1, Kim cannot teach or suggest each and every element in claim 13. The rejection of claim 13 over Kim is improper. Applicants therefore request that the rejection of claim 13, and of dependent claims 14-16, be withdrawn and claims 13-16 be allowed.

Claim Rejections - 35 USC § 103

The Action rejects claims 6-12 and 17 under 35 USC 103(a) as being unpatentable over a variety of references. In particular, the Action rejects independent claim 10 under 35 USC 103(a) as being unpatentable over Krumel, in view of Kim and Dugan. To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. (MPEP § 2142.)

Claim 10

Claim 10, as amended, recites:

upon the emulated device producing a packet of a type for which a defect behavior is represented in the defect configuration, and before transmitting the packet, applying the defect behavior to the packet; and

transmitting the packet from the emulated device as modified by applying the defect behavior.

[Emphasis added.] For example, the Application describes the actions of a "defect callback handler" that works as part of a device emulator:

All the packets before being sent out on the socket are consulted with the Defect Callback Handler object 342. . . . Each method will inject a defect or defects into a particular message type. *The user creates a defect behavior type by specifying a set of such methods using the defect configuration file 350. . . .* If no defect behavior is

> specified, the generic device emulator will emulate a perfect working device that is compliant with the UPnP™ Architecture 100.

[Application, at page 18, lines 10-17; emphasis added.] A particular example of the usefulness of the addition of defects, as well as a type of defect, is shown at page 14, line 27 to page 15, line 4:

> The generic device emulator 210 further provides a mechanism (described more fully below) for the vendor to define defective behaviors, which can be useful for testing control points 110-111 (Figure 1). *The defective behaviors are defined using XML format statements in a textual configuration file, which describe defect filters to be applied to the messages generated by the generic device emulator for the emulated device.* For example, a defect filter can be defined to have the generic device emulator 210 strip off a leading '*' character from headers of SOAP messages sent for the emulated device.

[Emphasis added.]

Krumel's filtering using a firewall cannot apply defect behavior "upon the emulated device producing a packet" and "before transmitting [a] packet" as recited in claim 10. Krumel describes "[m]ethods and systems for firewall/data protection" which work by "filter[ing] data packets in real time." [Krumel, at Abstract.] However, because Krumel describes a firewall, Krumel's filtering is performed by packets as they filter *through* the firewall:

> A packet filter is a device that examines network packet headers and related information, and determines whether the packet is allowed into or out of a network. . . .
>
> In accordance with the present invention, as the data of a packet comes in from one link (port), the packet's electrical signal is reshaped and then transmitted down other links. During this process, however, a filtering decision is made between the time the first bit is received on the incoming port and the time the last bit is transmitted on the outgoing links. During this short interval, a substantial number of filtering rules or checks are performed, resulting in a determination as to whether the packet should or should not be invalidated by the time that the last bit is transmitted.

[Krumel, at column 2, lines 24-50.] Thus, Krumel, by acting as a firewall, serves only to filter (and modify) packets which *are already passing through the firewall.*

Claim 10, by contrast, recites "upon the emulated device producing a packet . . . , and before transmitting the packet, applying the defect behavior to the packet," and "transmitting the packet from the emulated device as modified by applying the defect behavior." It would be impossible for the firewall described in Krumel to teach or suggest this language, however, because Krumel's firewall, by it's nature, only acts on packets that are sent to it, and to which it sends on. Thus, it would have no

opportunity, nor ability, to act "upon the emulated device producing a packet . . . and before transmitting the packet" "from the emulated device," as the packet has not been transmitted yet, and could not be received by the firewall.

For at least these reasons, Krumel does not teach or suggest the above-quoted language of claim 10. Furthermore, Applicants do not find such disclosure in either Kim or Dugan. Thus, the rejection of claim 10 over Krumel in view of Kim and Dugan is improper. Applicants request that the rejection of claim 10 be withdrawn.

Applicants also request that the rejections of claims 11 and 12, each of which depend from claim 10, be withdrawn. Claim 11 is rejected over identical art as claim 10, and claim 12 is additionally rejected over Chirashnya, from which Applicants do not find additional disclosure for the above-quoted language of claim 10. Thus, Applicants request that claims 10-12 be allowed.

Finally, the Action rejects claims 6-9 and 17 under 35 USC 103(a) as being unpatentable over Kim et al as applied to claims 1 and 13, respectively, and further in view of Chirashnya (claims 6, 8, and 9), Chirashnya and Krumel (claim 7), or Krumel and Dugan (claim 17). Because Applicants do not find additional disclosure for the above-quoted language of claims 1 and 13 in either Chirashnya, Krumel, or Dugan, Applicants respectfully submit that, for at least the reasons discussed above with respect to claims 1 and 13, the rejections of claims 6-9 and 17 are improper. Applicants request that the rejections of claims 6-9 and 17 be withdrawn and that the claims be allowed.

Request For Interview

If any issues remain in light of these remarks and amendments, the Examiner is formally requested to contact the undersigned attorney to arrange a telephonic interview. It is believed that a brief discussion of the merits of the present application may expedite prosecution. Applicants submit the preceding formal Amendment and the above remarks so that the Examiner may fully evaluate Applicants' position, thereby enabling the interview to be more focused.

This request is being submitted under MPEP § 713.01, which indicates that an interview may be arranged in advance by a written request.

Conclusion

The claims in their present form should be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 595-5300
Facsimile: (503) 595-5301

By _____
Stephen A. Wight
Registration No. 37,759